Abstract No: 814 ***IT, Mathematics and Statistics***

ON THE EFFICIENCY AND ACCURACY OF A FAST COMPUTATION OF DISCRETE MONOMIAL TRANSFORM

W.A. Gunarathna[1]* and H.M. Nasir[2]

[1]Department of IT and Mathematics, Faculty of Engineering,
General Sir John Kotelawala Defense University, Sri Lanka
[2]Department of Mathematics, Faculty of Science, University of Peradeniya, Sri Lanka
**gunarathnawa@yahoo.com*

Let $\{z_0, z_1, \ldots, z_{n-1}\}$ be a given set of n sample points. The n-point Discrete Monomial Transform (DMT) of an input vector $f = (f_0, f_1, \ldots, f_{n-1})$ is defined by the collection: $F = \{F(0), F(1), \ldots, F(n-1)\}$, where $F(k) = \sum_{j=0}^{n-1} f_j {z_k}^j$ for $0 \le k \le n-1$. The DMT is a widely used numerical tool in numerous scientific and technical applications, including Magnetic-resonance imaging (MRI), signal and image processing, weather forecasting, and engineering. The direct method (algo. 1) for computing the DMT at n sample points requires an $O(n^2)$ computational complexity. This cost gets prohibitively increased for large values of n and hence it makes a heavy burden in many DMT applications. In this paper, we consider an $O(n \log_2{}^2 n)$ algorithm (algo. 2) for computing an n point DMT. This algorithm takes advantage of the natural divide and conquer algorithm, fast polynomial multiplication via the Fast Fourier Transform (FFT), and fast Toeplitz matrix-vector multiplication. The key objective of this paper is to assess the accuracy and time efficiency of this algorithm when it is implemented in finite precision computer arithmetic.

MATLAB programs are coded using a tree data structure in MATLAB to implement algo. 2, and two numerical tests are carried out to test the performance of algo. 2. The CPU times taken by each algorithm are computed and compared for each numerical computation. The accuracy of the algorithm is computed by the absolute relative errors of the numerical solutions of the same numerical tests.

The experiment results demonstrate that algo.2 is more efficient than algo. 1 when $n \ge 128$. However, the absolute relative errors of the numerical solution show the accuracy of algo. 2 is drastically contaminated by lots of errors when $n > 16$. The tests done using matrix condition number indicate that the lower triangular matrices associated with algo. 2 become ill-conditioned when $n > 16$. Possible factors which affect the contamination of the accuracy of algo. 2 are identified.

Although algo. 2 is exact in hand computations, it can yield unreliable numerical errors for problems of modest size when it is implemented in reasonable computer arithmetic. However, its time efficiency can be trusted since it applies the same set of instructions for every problem size.